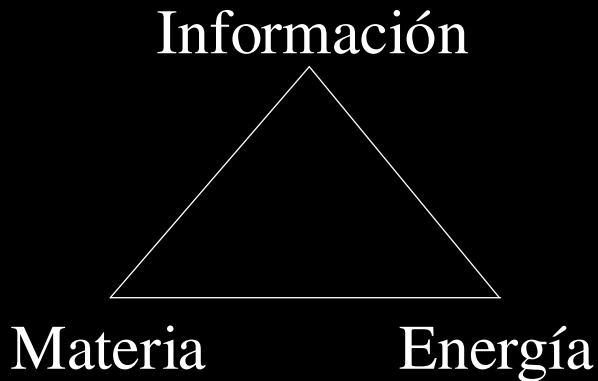


# *Desarrollo de productos de Informática*

[http://www.fgalindosoria.com/alta\\_direccion/d\\_i/2desarrollo\\_productos\\_Informatica.ppt](http://www.fgalindosoria.com/alta_direccion/d_i/2desarrollo_productos_Informatica.ppt)  
[http://www.fgalindosoria.com/alta\\_direccion/d\\_i/2desarrollo\\_productos\\_Informatica.pdf](http://www.fgalindosoria.com/alta_direccion/d_i/2desarrollo_productos_Informatica.pdf)



**REDI**

**Fernando Galindo Soria**

*fgalindo@ipn.mx*

*www.fgalindosoria.com*

**18 de Abril del 2014**

***MÉXICO***

*El mundo está cambiando rápidamente*

*La velocidad a la que surgen nuevas áreas es enorme,*

*y cada vez es más claro que una causa de la diferencia entre las personas, organizaciones y países ricos y pobres, es la diferencia entre personas, organizaciones y países rápidos y lentos.*

*Los lentos de ahora engrosarán el grupo de los pobres del futuro.*

*Taller técnico administrativo de dos días sobre  
Desarrollo de Productos de Informática*

*Miércoles 7, de 16:00 hrs. a 20:00 hrs.*

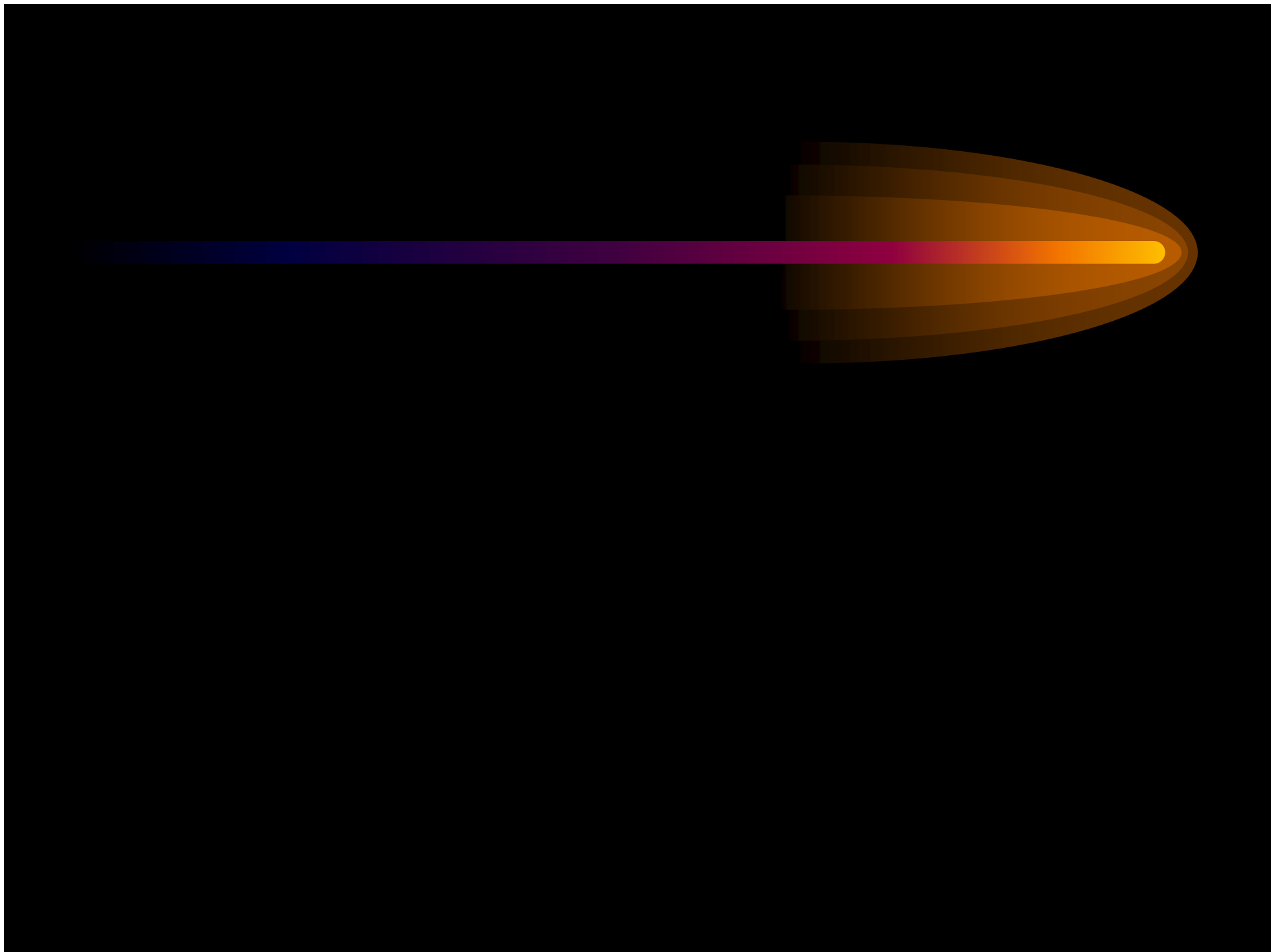
*Y Jueves 8, de 15:00 hrs. a 19:00 hrs*


*Se busca conocer los productos que están desarrollando  
los participantes a nivel industrial y académico,*

*y en su momento lograr que las instituciones académicas,  
empresas, organizaciones o personas interesadas lleguen  
a tener productos que compitan en el entorno mundial.*

*El eje del ciclo es el de: lograr que los participantes  
desarrollen industrias que compitan a nivel internacional  
con sus productos.*

*Conocer y apoyar el desarrollo de  
los productos de los participantes*





*lograr que las instituciones académicas,  
empresas, organizaciones o personas  
interesadas lleguen a tener productos que  
compitan en el entorno mundial.*

# *Características de un producto*

*Eficaz,*

*Eficiente*

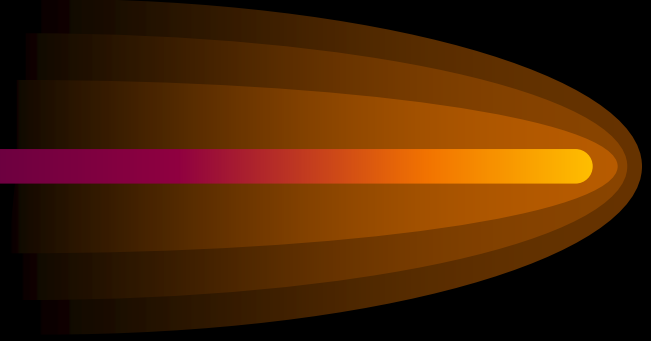
*Amigable,*

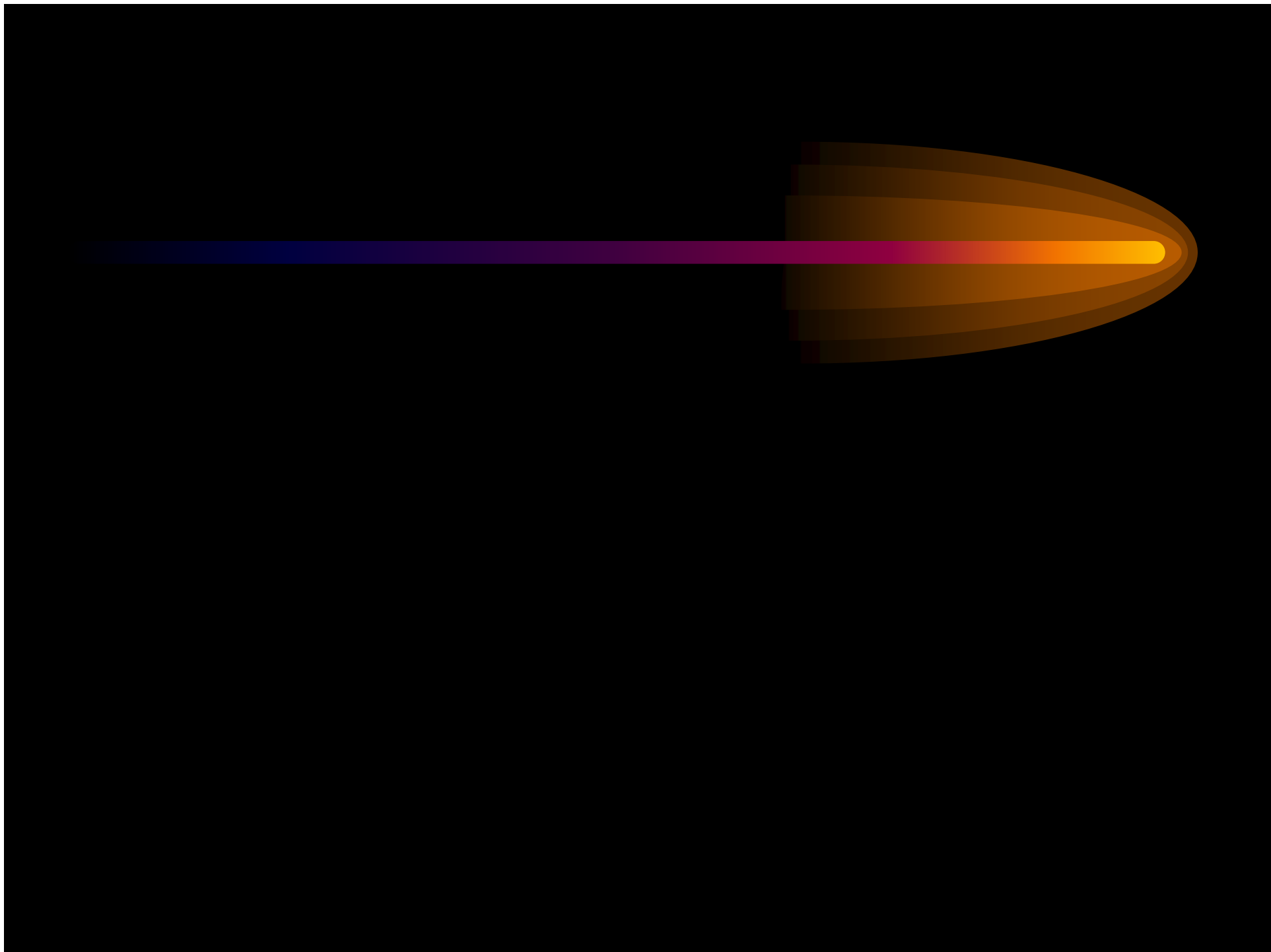
*Independiente del usuario,*

*Parametrizado,*

*presentación, empaque*

*Robusto*







# *Características de un producto*

*Eficaz,*

*Eficiente*

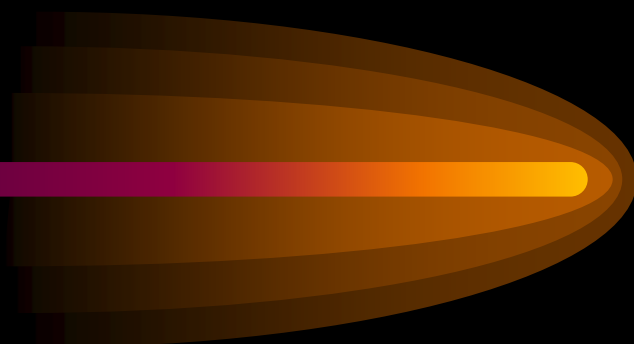
*Amigable,*

*Independiente del usuario,*

*Parametrizado,*

*presentación, empaque*

*Robusto*



*Cuando algo se hace varias veces, llega un momento que se detecta que no tiene sentido seguir repitiendo lo mismo*

*Si se hace algo varias veces conviene delegar la actividad (se les explica a otros como hacerlo y se les delega)*

*o se hace un programa que lo haga*

*O se hace un generador de programas*

*Cuando algo se hace varias veces, llega un momento que se detecta que no tiene sentido seguir repitiendo lo mismo*

*Si se hace algo varias veces conviene delegar la actividad (se les explica a otros como hacerlo y se les delega)*

*o se hace un programa que lo haga*

*O se hace un generador de programas*

*Generador de Programas*

*Fernando Galindo Soria*

*1975 – 1976*



*Generador de Programas*

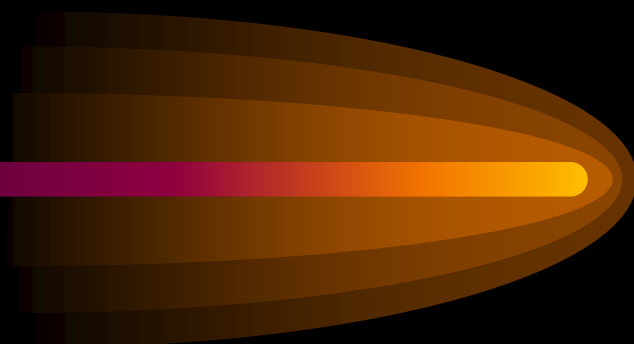
*Fernando Galindo Soria*


*1975 – 1976*

*Generador de Programas*

*Vicente López Trueba*

*1980*





*Cuando algo se hace varias veces, llega un momento que se detecta que existen cosas que se repiten y cosas que cambian de una versión a la siguiente*

# *Generador de Programas*

*Se toma un programa, por ejemplo el de la suma*

*Lee a, b*

*$c=a+b$*

*Escribe c*



# *Generador de Programas*

*Se toma un programa, por ejemplo el de la suma*

*Lee a, b*

*$c=a+b$*

*Escribe c*

*Se toma otro programa también de la suma*

*Lee x, y*

*$z=x+y$*

*Escribe z*



*Se comparan los programas y se detecta lo que cambia*

*Lee a, b*

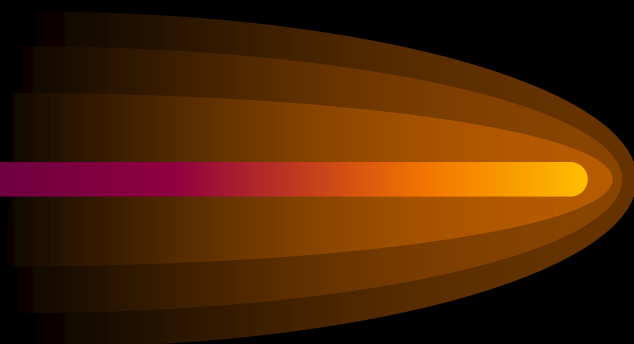
*Lee x, y*

*c=a+b*

*z=x+y*

*Escribe c*

*Escribe z*



*Se comparan los programas y se detecta lo que cambia*

*Lee **a**, **b***

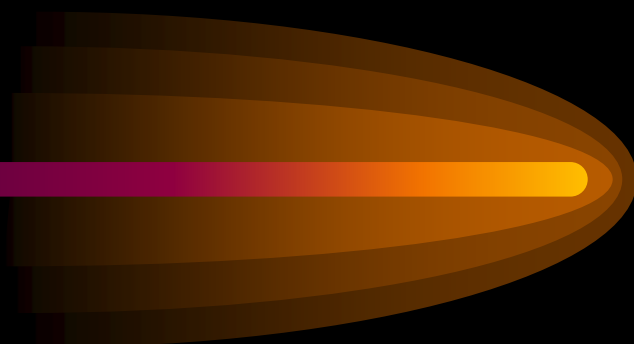
*Lee **x**, **y***

***c**=**a**+**b***

***z**=**x**+**y***

*Escribe **c***

*Escribe **z***



*Se comparan los programas y se detecta lo que cambia*

*Lee **a**, **b**                      Lee **x**, **y***

***c**=**a**+**b**                              **z**=**x**+**y***

*Escribe **c**                              Escribe **z***

*Se sustituyen los elementos que cambian en el programa por una etiqueta **\$1**, **\$2**, **S3***

*Lee **\$1**, **\$2***

***S3**=**\$1**+**\$2***

*Escribe **\$3***

*A los elementos que cambian se les llama **parámetros***

*Estamos usando el símbolo \$ para indicar parámetro, pero se puede usar cualquier simbolo que no se use dentro del sistema (como #, “, etc.)*

*Se comparan los programas y se detecta lo que cambia*

*Lee **a**, **b**                      Lee **x**, **y***

***c**=**a**+**b**                              **z**=**x**+**y***

*Escribe **c**                              Escribe **z***

*Se sustituyen los elementos que cambian en el programa por una etiqueta **\$1**, **\$2**, **S3***

*Lee **\$1**, **\$2***

***S3**=**\$1**+**\$2**                      esqueleto*

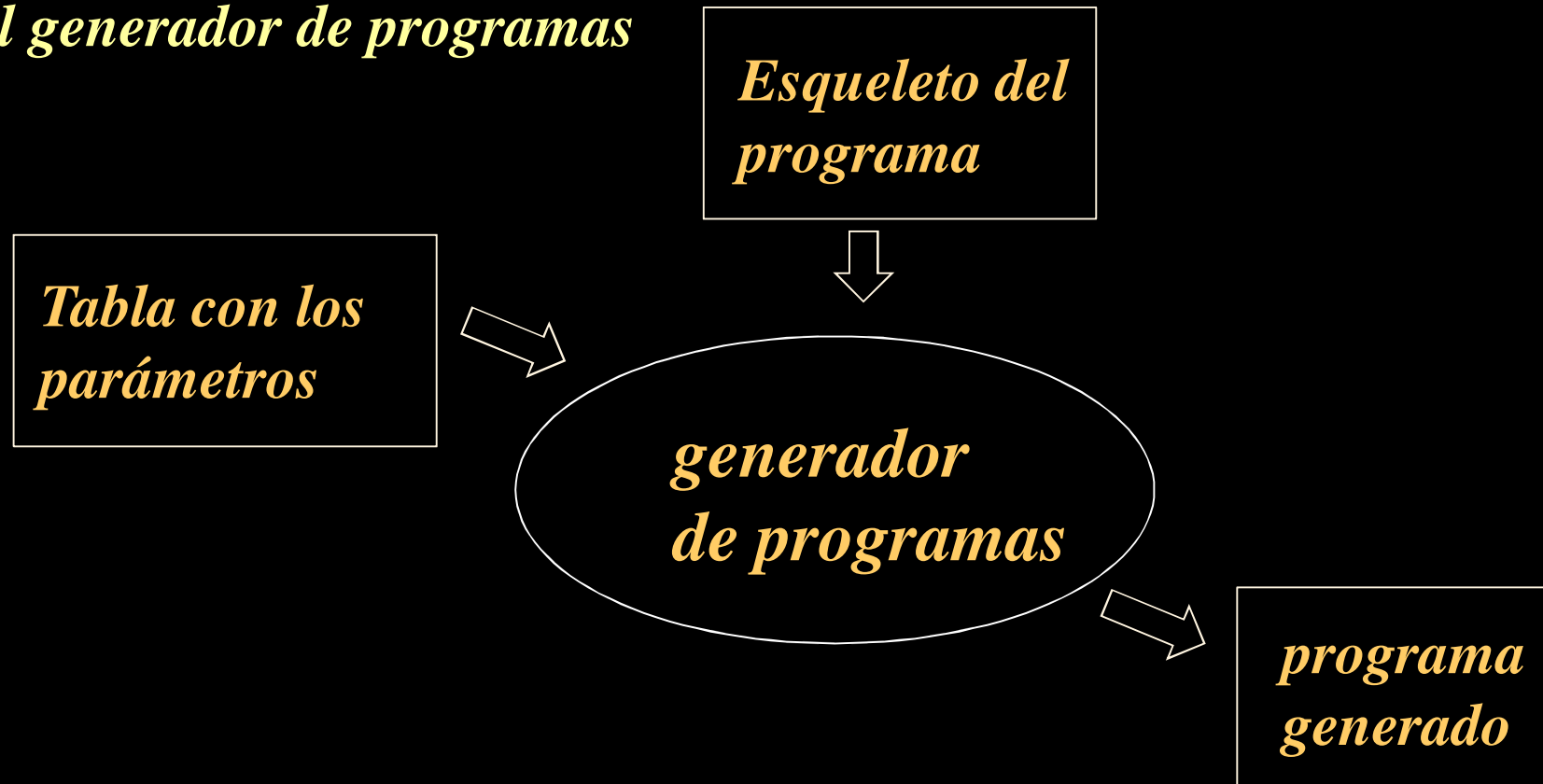
*Escribe **\$3***

*A los elementos que cambian se les llama parámetros*

*al programa con las etiquetas se le llama esqueleto y a este proceso se le llama **Parametrización***

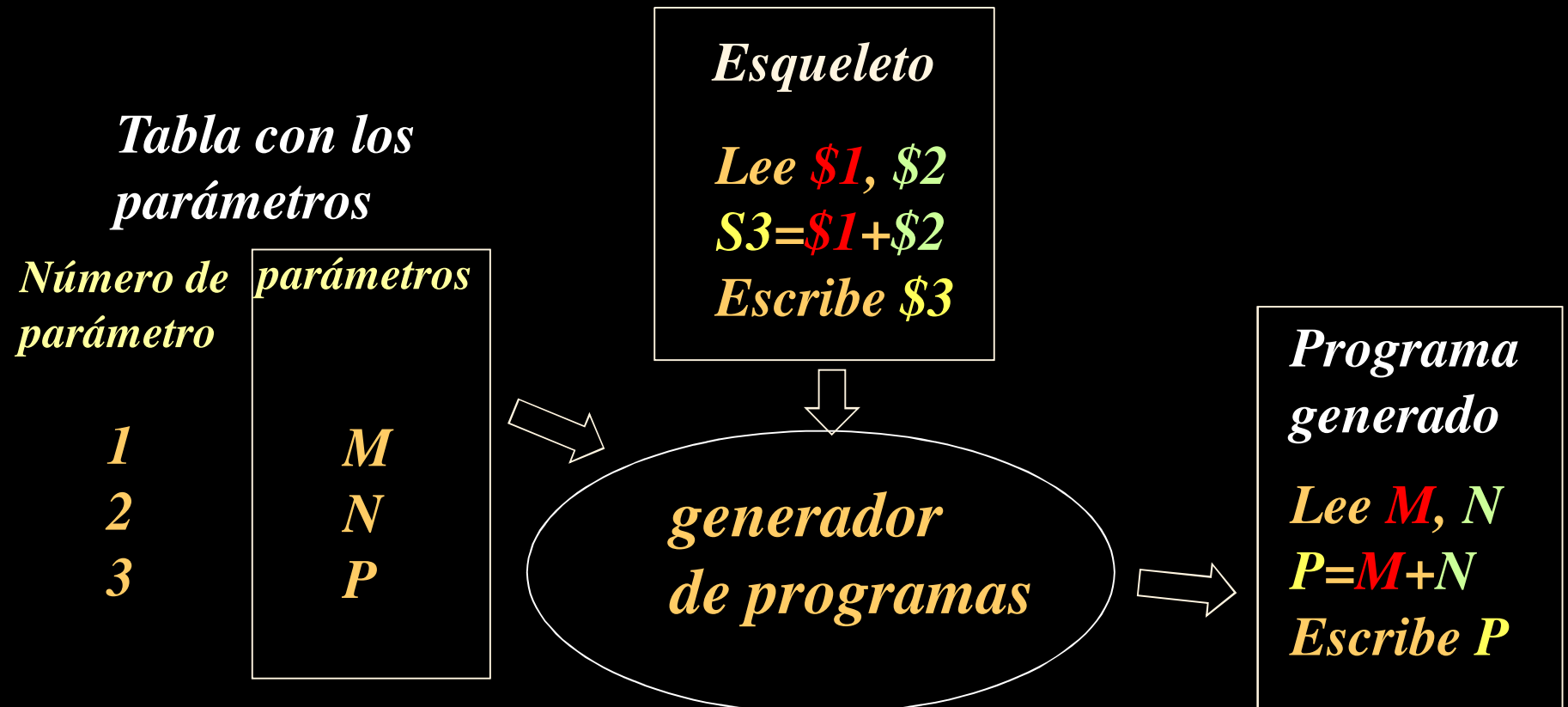
# *Generador de Programas*

*El sistema generador de programas consta de 4 componentes:  
el esqueleto del programa que se quiere generar,  
una tabla con los parámetros que se van a usar en este caso  
particular  
un archivo donde se almacena el programa generado  
y el generador de programas*



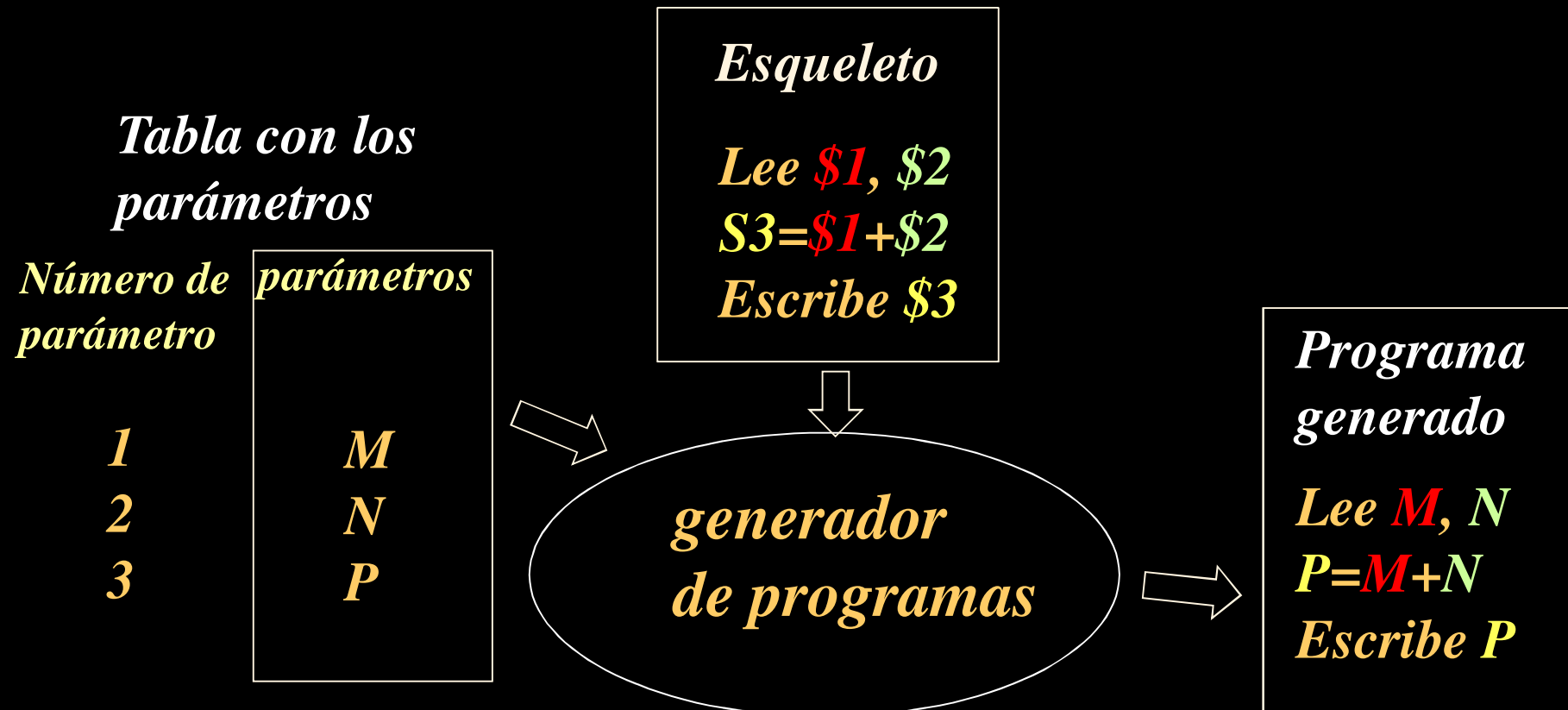
# Generador de Programas

Por ejemplo si se quiere hacer un programa que sume  $M$  con  $N$  y ponga el resultado en  $P$ , se colocan en la tabla de parámetros  $M$ ,  $N$  y  $P$ , se toma el esqueleto de la suma y se llama al generador de programas



# Generador de Programas

*El generador de programas comienza a leer el código que aparece en el esqueleto y si no es un parámetro lo reescribe en la tabla del Programa generado, pero si es un parámetro (o sea que tiene el carácter \$ entonces lee el número del parámetro, va a la tabla y toma ese parámetro y lo escribe en el programa generado)*



# Generador de Programas

## Esqueleto

Lee \$1, \$2  
S3=\$1+\$2  
Escribe \$3

## Tabla con los parámetros

Número de parámetro

Número de parámetro	parámetros
1	M
2	N
3	P

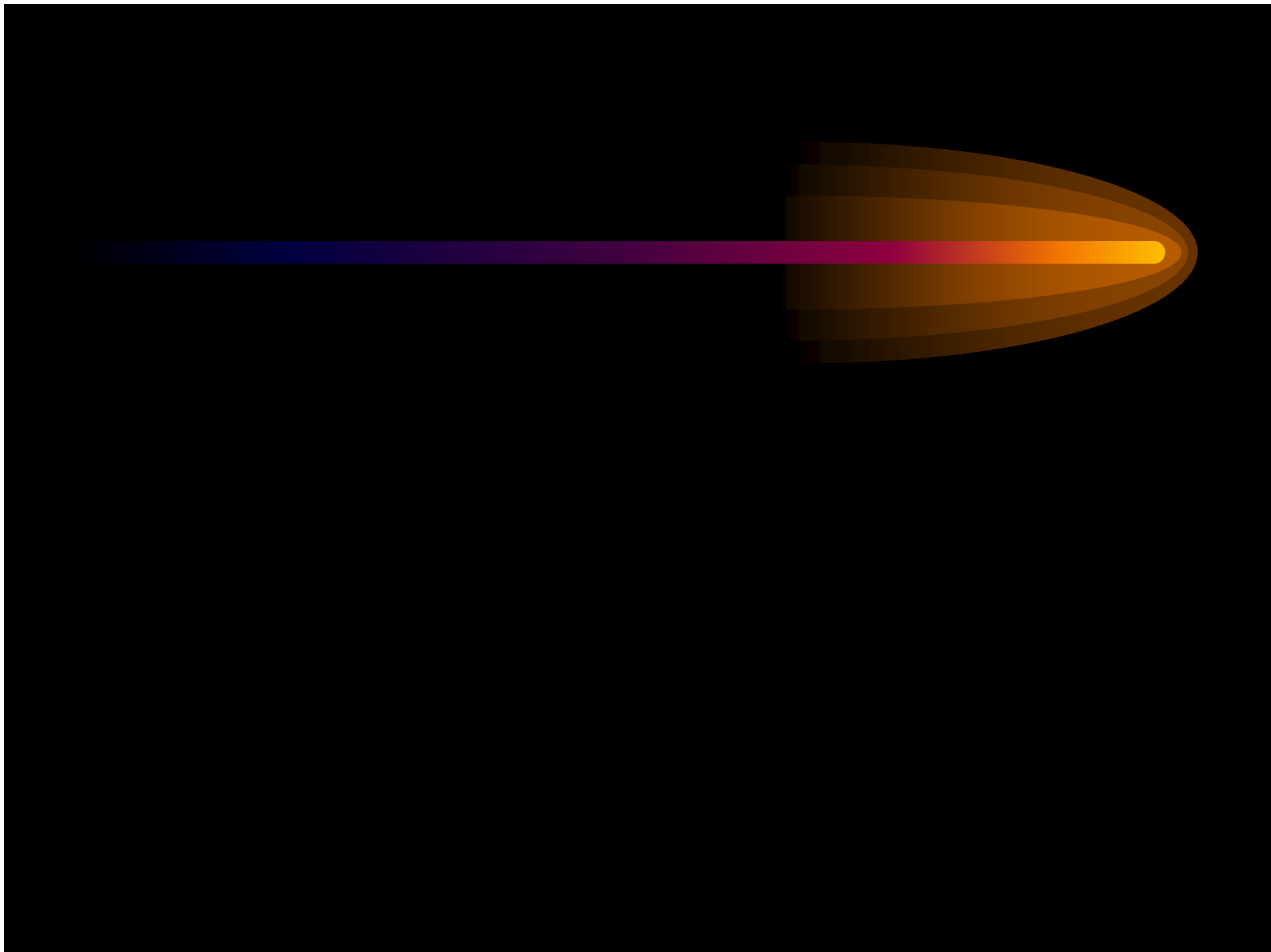
Lee siguiente carácter del Esqueleto  
Si no es parámetro (si no es el carácter \$)  
Escribe el carácter en Programa Generado  
Si es parámetro (si es el carácter \$)  
Lee el número de parámetro del Esqueleto  
Toma el parámetro de la Tabla de parámetros  
Escribe el parámetro en Programa Generado

Mientras no fin de archivo Repite el ciclo

## Programa generado

Lee M, N  
P=M+N  
Escribe P







*Generadores de programas*

*Generadores de sistemas*

*Sistemas evolutivos*

*Generador de Programas*

*Generador de Sistemas*

*Vicente López Trueba*

*1980*

*A partir de un Generador de Programas es relativamente fácil construir un Generador de Sistemas*

*En principio podemos ver un sistema como un conjunto de programas interrelacionados, por lo que para construir un generador de sistemas, lo primero que se hace es detectar los programas que forman el sistema  
cada programa se parametriza y se obtiene su esqueleto,*

*luego mientras existan programas a generar, el generador de sistemas llama al generador de programas que llama a cada programa y le asocia los datos particulares del sistema que se quiere generar*

*El único detalle que falta es como indicarle al generador de sistemas cual es el siguiente programa a generar*



*Generador de Programas*

*Generador de Sistemas*

*Vectores Teóricos*

*Vicente López Trueba*

*1980*

## *Vectores Teóricos*

*En una versión muy preliminar, un vector teórico es una lista de los programas que forman el sistema en el orden en que se deben generar*

*Por ejemplo si el sistema esta formado por los programas*

*A (pantalla de captura),*

*B (validación de datos),*

*C (calculo de resultados),*

*D (presentación de resultados)*

*El Vector Teórico quedaría como la siguiente tabla*

A	B	C	D
---	---	---	---

# Generador de Sistemas

Mientras existan programas en el vector teórico

Llama al generador de programas para que genere el siguiente programa

**Generador de Programas**

Lee siguiente carácter del Esqueleto

Si no es parámetro (si no es el carácter \$)

Escribe el carácter en Programa Generado

Si es parámetro (si es el carácter \$)

Lee el número de parámetro del Esqueleto

Toma el parámetro de la Tabla de parámetros

Escribe el parámetro en Programa Generado

Mientras no fin de archivo Repite el ciclo

Esqueleto  
Esqueleto  
Esqueleto  
Esqueleto

Lee \$1, \$2  
S3=\$1+\$2  
Escribe \$3

Tabla con  
los  
parámetros  
los  
parámetros  
parámetros

1	M
2	N
3	P

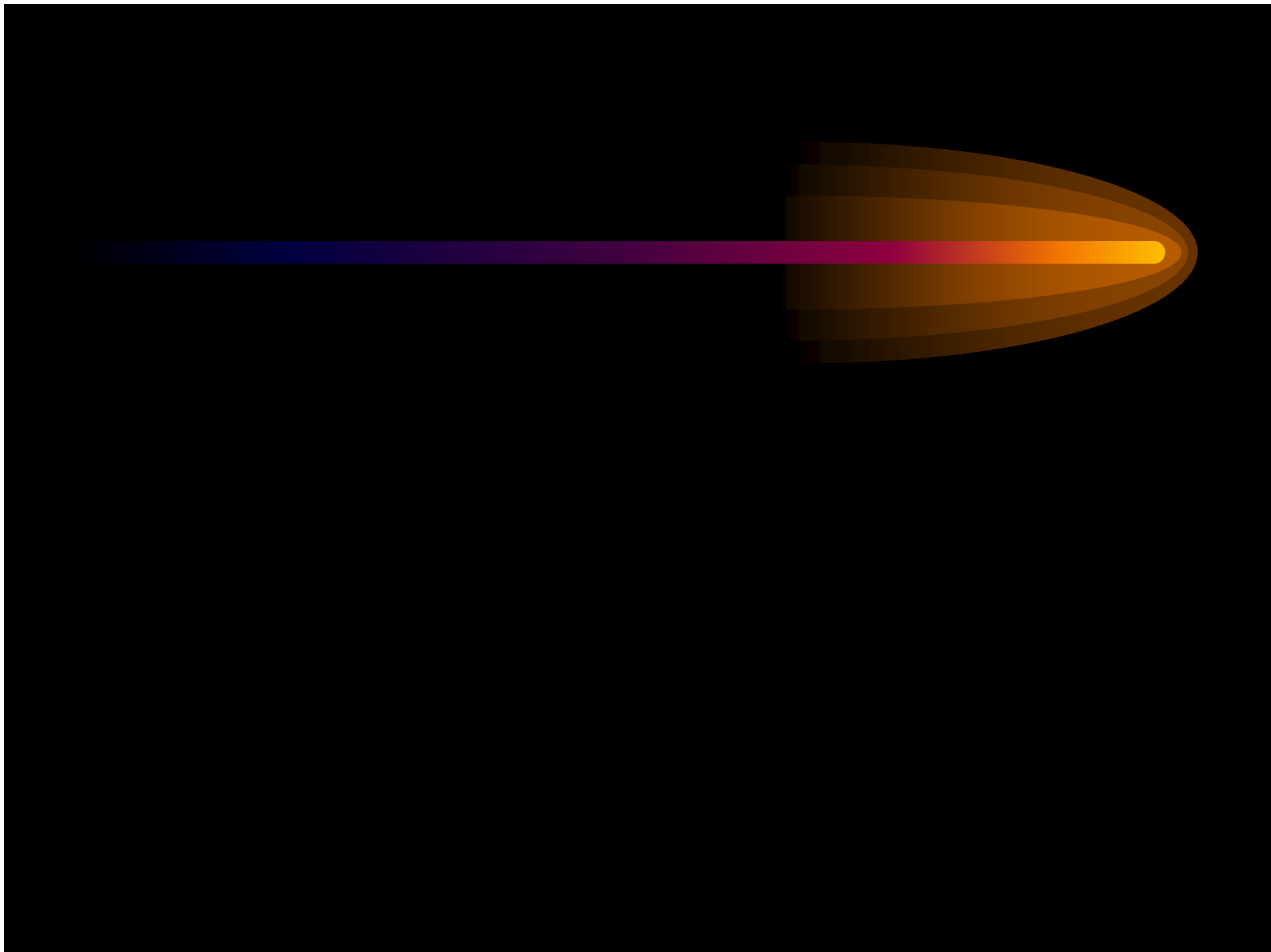
Vector Teórico



Sistema  
generado

A  
B  
C  
D





*Generador de Programas*

*Generador de Sistemas*

*Vectores Teóricos*

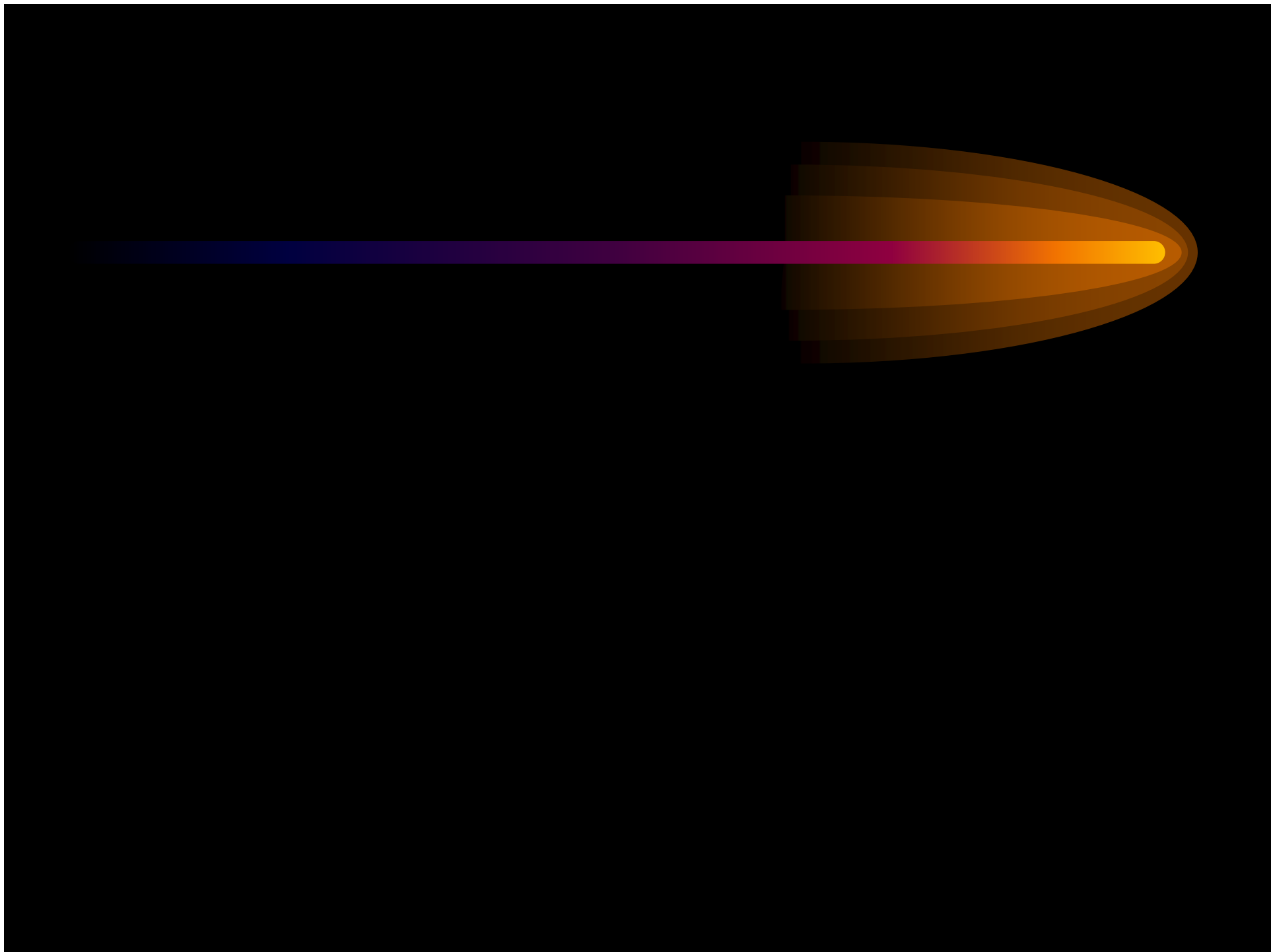
*Gramáticas*

*Gramática*



*Representación lingüística de la  
estructura de un lenguaje*

*Enfoque Lingüístico y Lingüística Matemática*  
<http://www.fgalindosoria.com/lingüisticamatematica/index.htm>



*Un sistema tiene datos, procesos y estructura*

*Vicente López Trueba*

*1980*



*Un sistema tiene datos, procesos y estructura*

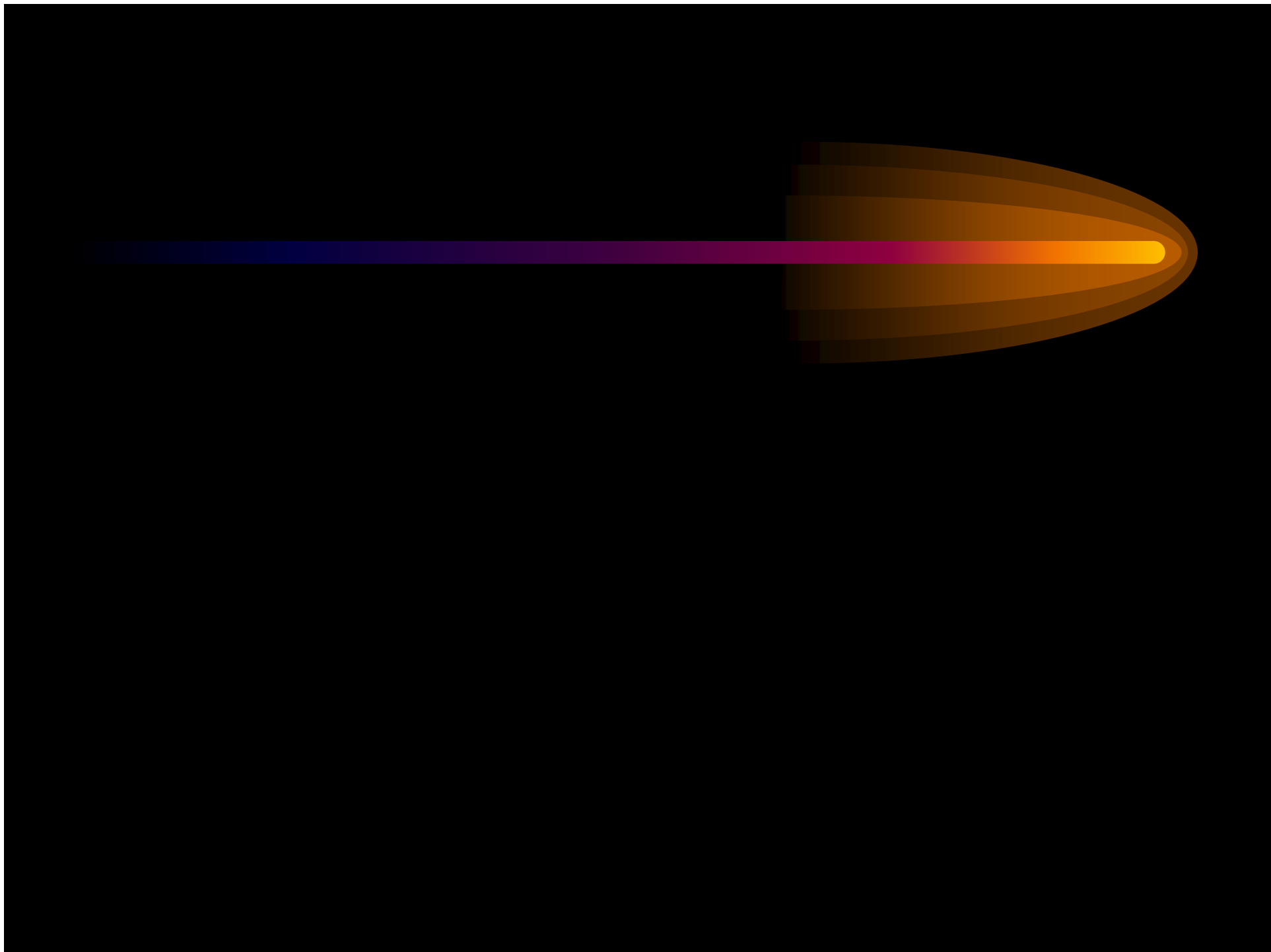
*Vicente López Trueba*

*1980*

*Datos*       $\Rightarrow$       *Parámetros*

*procesos*       $\Rightarrow$       *esqueletos*

*estructura*       $\Rightarrow$       *Vectores Teóricos*  
*Gramáticas*





*INDEPENDENCIA RELATIVA*

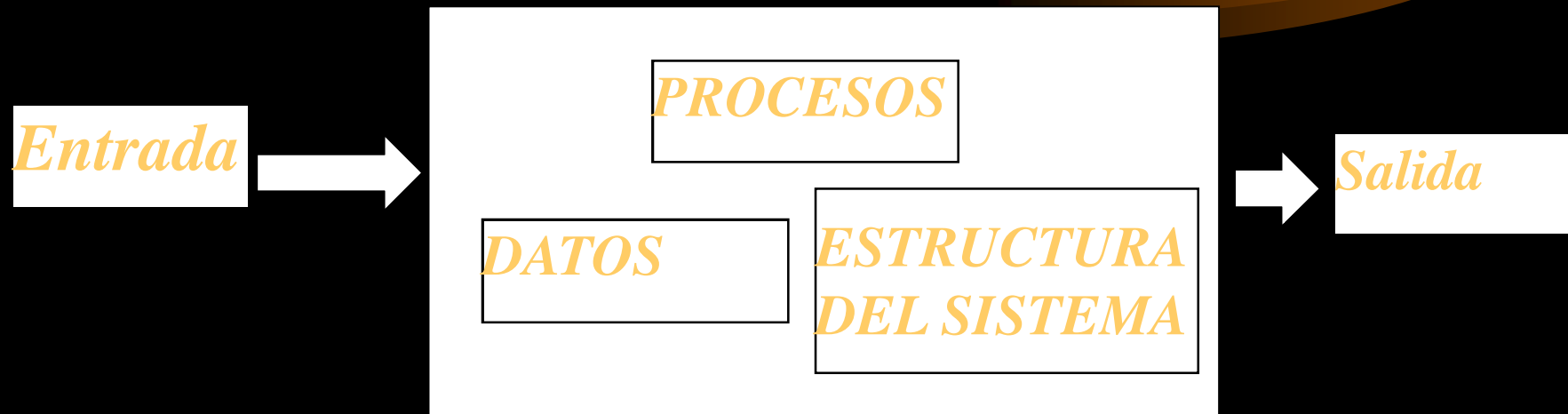


*En general se considera que un Sistema de Información o programa de cómputo tiene la arquitectura de la figura*



*Arquitectura de los sistemas de información en los 50s*

*ya para finales de los 70's se manejaba un modelo generalizado en el cual se considera que cualquier sistema de información tiene la arquitectura de la siguiente figura*



*Arquitectura de los sistemas de información en los 70s*

*El desconocimiento o el hecho de no tomar en cuenta este esquema cuando se desarrollan sistemas es una de las causas principales por las que los sistemas se vuelven altamente estáticos y difíciles de mantener*

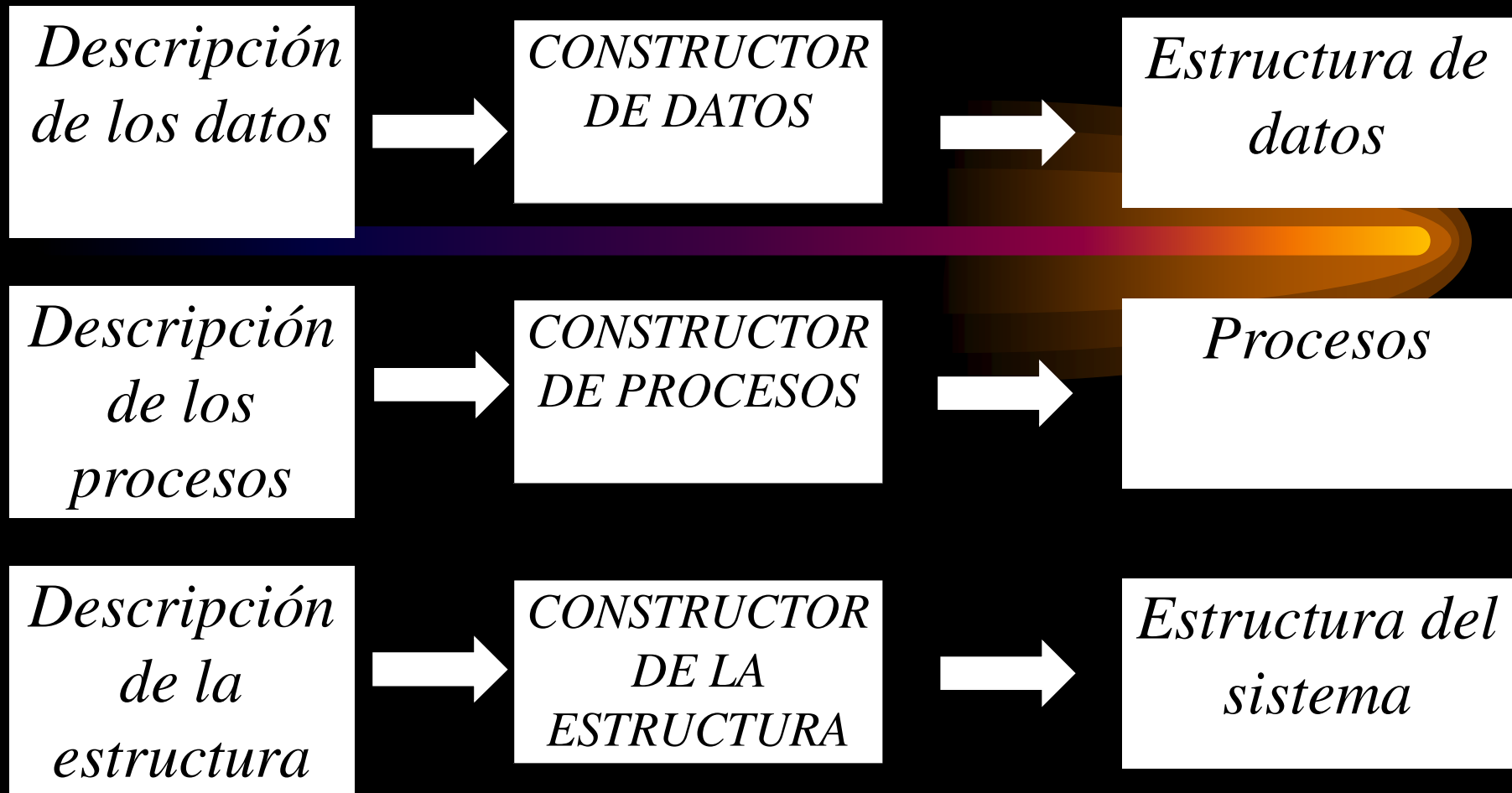
*ya que en los sistemas y programas tradicionales las tres componentes se encuentran revueltas, por lo que, un cambio "pequeño" en los datos o procesos o en el orden de atacar un problema ocasiona que prácticamente se tenga que volver a programar todo;*

*por otro lado si se desarrolla el sistema de tal manera que los datos queden en un lado, los procesos en otro y finalmente la estructura del sistema en otro, el proceso de actualización puede ser relativamente fácil (un caso particular de este enfoque es el del desarrollo de Bases de Datos).*

*una característica fundamental que se debe buscar cuando se desarrolla un sistema de información es la de que exista una*

## *Independencia Relativa*

*los datos, procesos y estructura del sistema queden separados y únicamente existe la relación mínima necesaria entre las tres componentes.*



***Constructores de datos, procesos y estructura***

*Independencia relativa entre datos,  
procesos y estructura*



*Sistemas evolutivos*

*Generador de Programas*

*Generador de Sistemas*

*Vectores Teóricos Gramáticas*

*Independencia Relativa*

*Sistemas Evolutivos*

*El sistema se transforma permanentemente a partir de los flujos de Materia, Energía e Información en los que esta inmerso*

<http://www.fgalindosoria.com/eac/evolucion/>

[http://www.fgalindosoria.com/eac/evolucion/evolucion\\_sev/evolucion\\_sev.pdf](http://www.fgalindosoria.com/eac/evolucion/evolucion_sev/evolucion_sev.pdf)

[http://www.fgalindosoria.com/eac/evolucion/evolucion\\_sev/evolucion\\_sev\\_a.pdf](http://www.fgalindosoria.com/eac/evolucion/evolucion_sev/evolucion_sev_a.pdf)

*Vicente López Trueba*

*Christian Zempoaltecatl Ibarra*

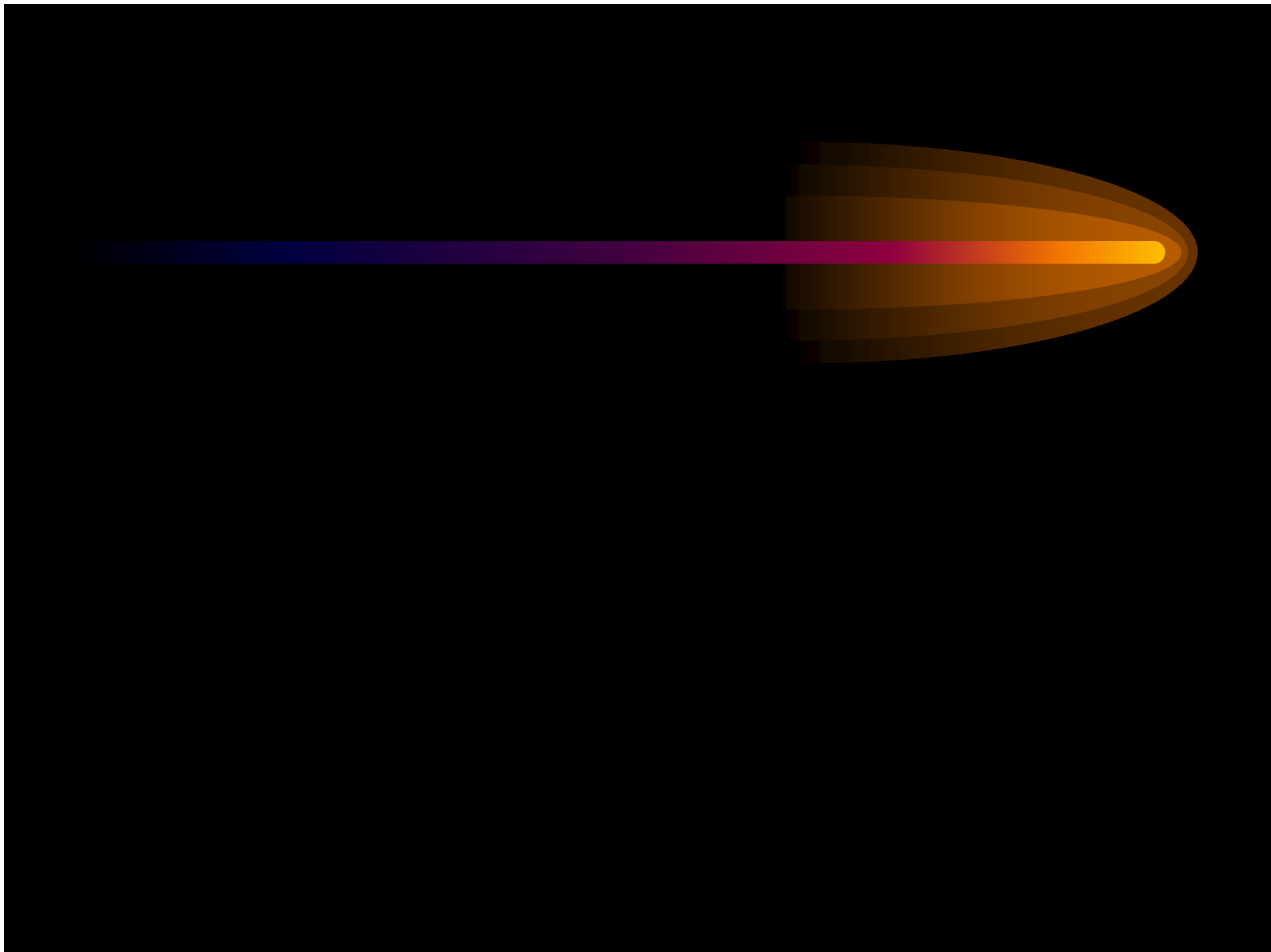
*Ricardo García*

*Gabriel Cordero*

*Juan Martín González Vázquez*

*Cuitláhuac Cantú Rohlík*





*Como transformar una tarea,  
programa o sistema en un producto*



*Tomar el sistema y parametrizarlo*

## *Parametrización*

*Se hacen varias versiones del sistema, hasta que se tengan claro cuales son las cosas que se conservan y cuales cambian de una versión a otra*

*Por ejemplo en el programa que calcula probabilidades, en general lo que cambia es el nombre del archivo donde están los datos y cuantos datos son*

## *Se construye la interfase de usuario*

*por ejemplo en el caso del programa que calcula probabilidades, la interfase presenta una pantalla al usuario donde pide los parámetros, en este caso cuantos datos son y el nombre del archivo donde están los datos.*

*La interfase pide la información, la recibe, llama a la rutina que calcula probabilidades y entrega el resultado al usuario.*

*Muchos productos de mercado tiene la estructura anterior, por lo que está es la base de la construcción de muchos productos.*

*Ahora bien muchos sistemas tradicionales tiene también esta estructura, por lo que es relativamente fácil cambiar muchos sistemas tradicionales en productos.*

*Por ejemplo, en muchos sistemas solo es necesario ver lo que cambia del sistema que se hace para una empresa y el que se hace para otra empresa, por ejemplo algo de lo que cambia son los datos de la empresa a la que se le hace el sistema (nombre, dirección, razón social, etc.),*

*cuando se parametriza el sistema lo normal es que la información general se almacena en alguna tabla y cuando arranca el sistema se toma esa información de la tabla (en varios productos del mercado la función principal de los analista del sistema es construir esas tabla)*

*Se toma el sistema que se quiere generalizar*

*Se detecta los parámetros (los datos que cambian de una aplicación a otra: nombre de la empresa, dirección, número de campos, ubicación de los archivos, etc.)*

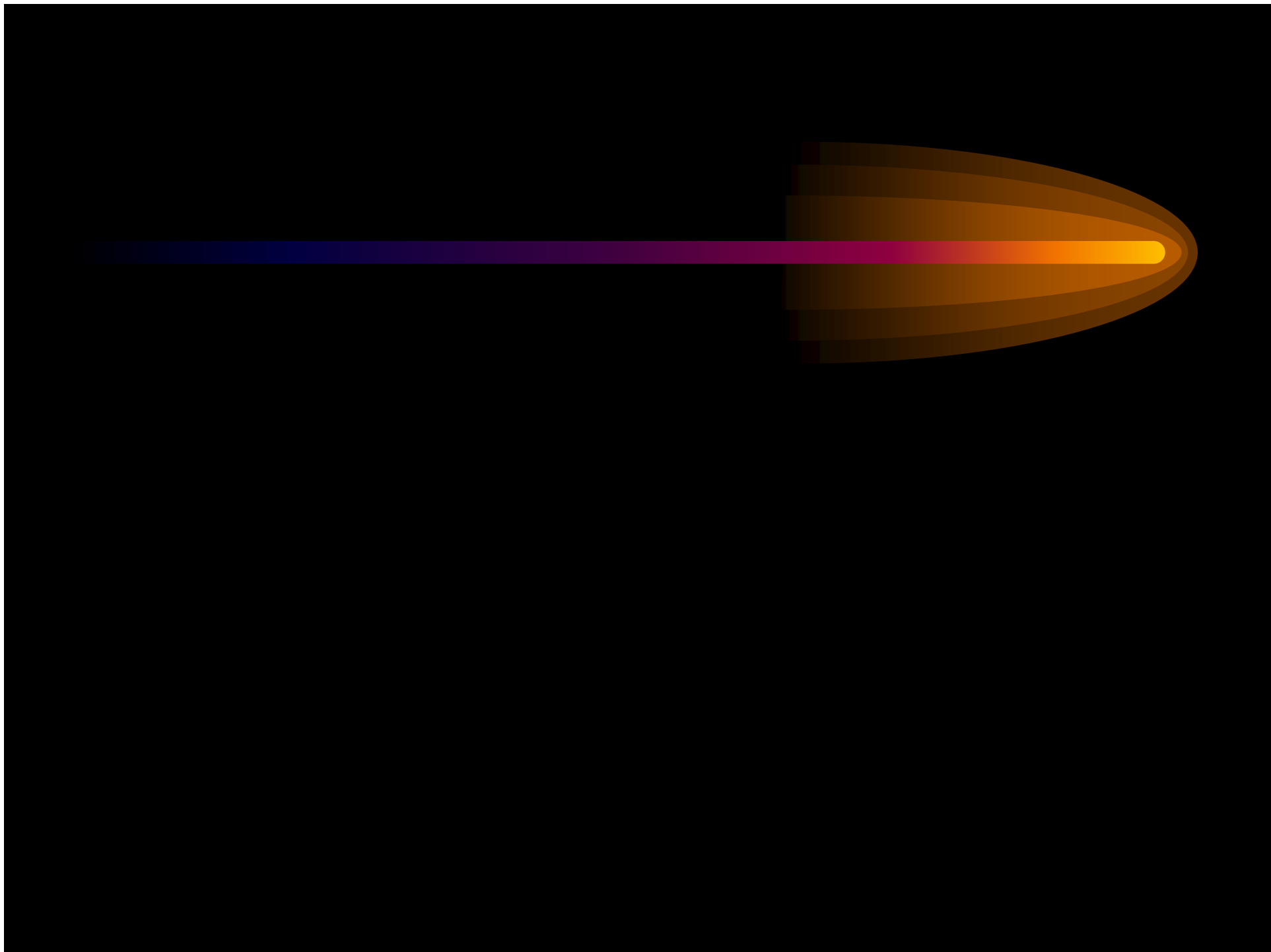
*Se construye el esqueleto a partir del sistema*

*Se construye la interface del usuario que:*

*pide la información que cambia al usuario (nombre de la empresa, dirección, número de campos, ubicación de los archivos, etc.)*

*la recibe, rellena los campos particulares del sistema y entrega el resultado al usuario.*

*Se instala y usa el sistema particularizado*



# *Características de un producto*

*Eficaz,*

*Eficiente*

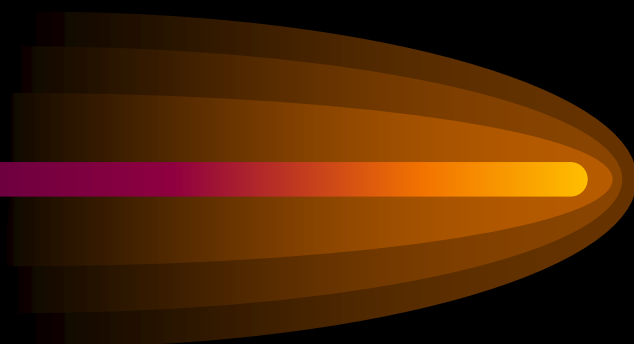
*Amigable,*

*Independiente del usuario,*

*Parametrizado,*

*presentación, empaque*

*Robusto*







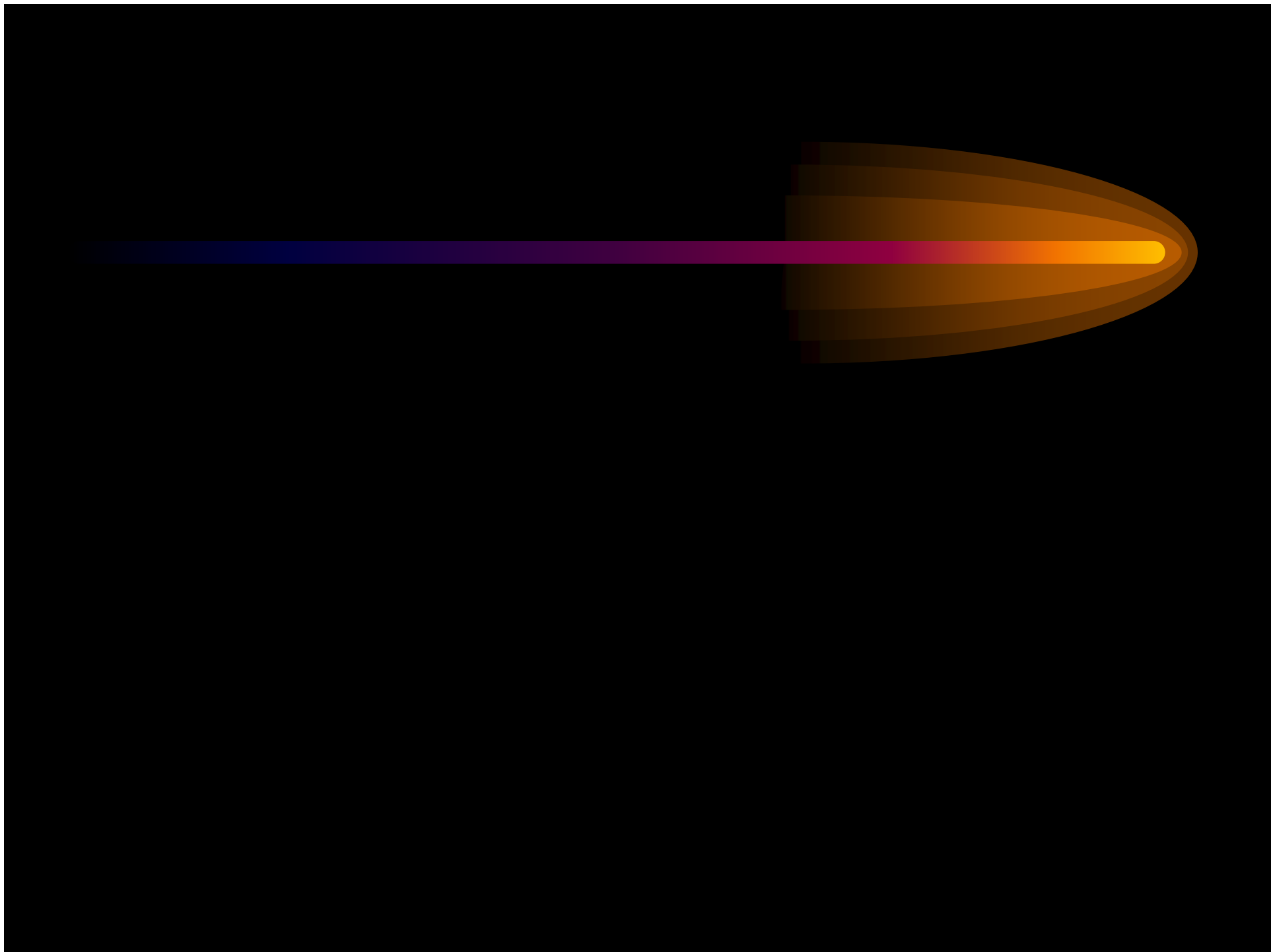
*La diferencia entre una tarea y un  
producto es el empaque  
Empaque, presentación*

*Otra actividad dentro del desarrollo de un producto es darle Presentación de Producto (empaquete)*

*por ejemplo Nombre, pantalla de entrada, pantalla de fin, etc.*

*Algunos productos requieren también un empaque físico (dvd, frasco, caja), etiquetas, empaques, etc., pero en general, todo esto se consigue relativamente fácil*

*Software (dvd, red), Lociones (frascos, etiqueta)  
Circuitos (cajas), Robots (perros, carros), etc.*



*Algunas áreas de la Informática Administrativa  
Sociedad de la Información*

*Comercio en Red, nube, Industria del  
Conocimiento*

*Gobierno Electrónico*

*Penetración de la red (zonas marginales, zonas de  
difícil acceso)*

*Manejo masivo de datos (Big Data / Grandes  
volúmenes de datos, Data Mining / minería de  
datos)*

*Contestador automático (call center)*

*Manejo de almacenes*

*Sistemas de apoyo a la toma de decisiones*

*Varios niveles de ataque*

*Usar la tecnología como caja negra*

*Analizar los problemas e integrar la tecnología para resolverlos*

*Desarrollar las nuevas plataformas, herramientas, productos*

*Se requiere el dominio de áreas como:*

*ICU*

*Interactividad.*

*Industria del Conocimiento*

*Informática Ubicua*

*Computación*

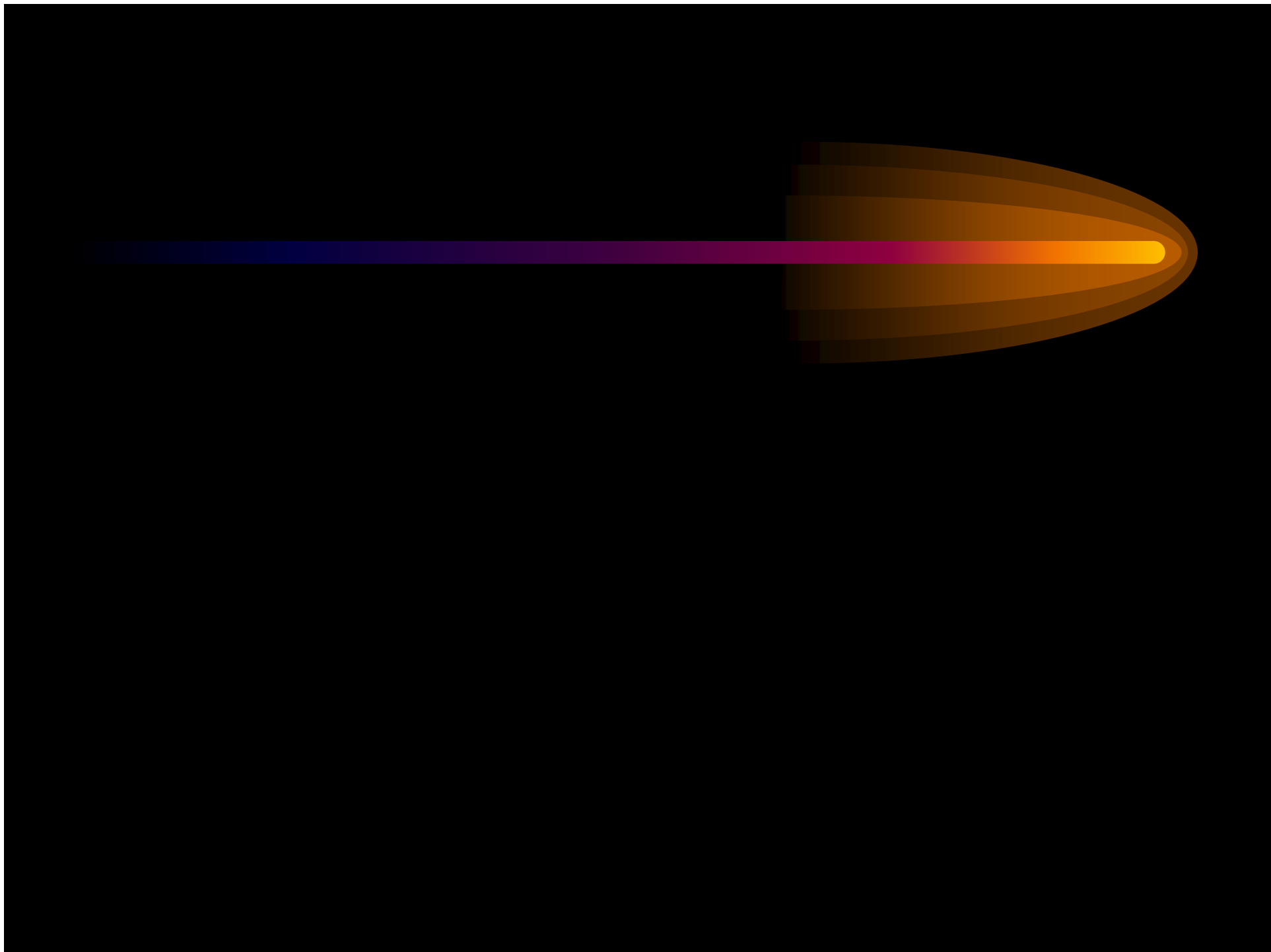
*Comunicaciones*

*Inteligencia Artificial.*

*Robótica y automatización.*

*Reconocimiento de patrones.*

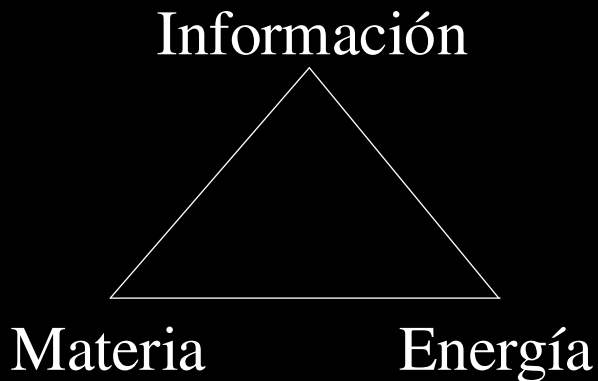
*Etc.*



# *Desarrollo de productos de Informática*

[http://www.fgalindosoria.com/alta\\_direccion/d\\_i/2desarrollo\\_productos\\_Informatica.ppt](http://www.fgalindosoria.com/alta_direccion/d_i/2desarrollo_productos_Informatica.ppt)

[http://www.fgalindosoria.com/alta\\_direccion/d\\_i/2desarrollo\\_productos\\_Informatica.pdf](http://www.fgalindosoria.com/alta_direccion/d_i/2desarrollo_productos_Informatica.pdf)



**REDI**

**Fernando Galindo Soria**

*fgalindo@ipn.mx*

*www.fgalindosoria.com*

**18 de Abril del 2014**

***MÉXICO***